Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 648–655

Procedia Computer Science

www.elsevier.com/locate/procedia

The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40), April 6 - 9, 2020, Warsaw, Poland

Architecture Proposal for Machine Learning Based Industrial Process Monitoring

Lorenz Rychener*[a], Frédéric Montet[a], Jean Hennebert[a]

[a]*Institute of Complex Systems, HES-SO, HEIA-FR, Boulevard de Pérolles, 1700 Fribourg, Switzerland*

Abstract

In the context of Industry 4.0, an emerging trend is to increase the reliability of industrial process by using machine learning (ML) to detect anomalies of production machines. The main advantages of ML are in the ability to (1) capture non-linear phenomena, (2) adapt to many different processes without human intervention and (3) learn incrementally and improve over time. In this paper, we take the perspective of IT system architects and analyse the implications of the inclusion of ML components into a traditional anomaly detection systems. Through a prototype that we deployed for chemical reactors, our findings are that such ML components are impacting drastically the architecture of classical alarm systems. First, there is a need for long-term storage of the data that are used to train the models. Second, the training and usage of ML models can be CPU intensive and may request using specific resources. Third, there is no single algorithm that can detect machine errors. Fourth, human crafted alarm rules can now also include a learning process to improve these rules, for example by using active learning with a human-in-the-loop approach. These reasons are the motivations behind a microservice-based architecture for an alarm system in industrial machinery.

© 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the Conference Program Chairs.

Keywords: System Architecture; Rule Engine; Anomaly Detection; Monitoring; Industry 4.0

1. Introduction

Today's competitive industrial landscape is pushing industries towards shorter product development life-cycle and better production efficiency. These objectives are implying increased automation and monitoring of production machines, driving towards the Industry 4.0 [7]. In this context, more reliable anomaly detection systems are needed to orchestrate industrial machines.

* Corresponding author. Tel.: +41 26 429 65 67
E-mail address: lorenz.rychener@hefr.ch

1877-0509 © 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2020.03.137

Anomaly Detection (AD) has to deal with a plethora of different issues. The most prominent are: the existence of different categories of anomalies (point, contextual, collective), the definition of the normal behaviour of the system and its allowed evolution, the dependency to a domain, the rarity of examples of anomalies and the differentiation from noise [3].

Machine learning (ML) and more recently deep learning (DL) enjoy increasing popularity for anomaly detection. The main advantages of such data-driven systems are their ability to capture non-linear phenomena, adapt to many different processes without human intervention, learn incrementally and improve over time [2, 17, 6].

Different application cases of monitoring industrial machinery have been explored such as in power generation systems [4], oil industrial machinery like motors turbines and pipelines [19], vehicle and aircraft engines [18] and production, eg. slitting-, machines [9]. Closely related to AD, some work has been done to address the detection of fraudulent behaviour or malicious attacks on Cyber-Physical Systems (CPS) [5] in addition to automatic sensor fault detection [20].

In this paper, we present an approach to solve the challenges induced by the integration of ML into the world of industrial machinery. This approach starts with a review of the challenges caused by the monitoring of industrial processes. Then, we discuss the integration of machine learning into an industrial process. Finally, the architecture of an alarm system that combines classical and machine learning approaches to fault detection is proposed. The different parts of the architecture are designed to fit into a microservices running on a container orchestrator.

2. Industrial Process Monitoring

In the paradigm of Industry 4.0, physical and software components are often intertwined and form a so-called cyber-physical system [1]. Monitoring such systems usually brings some challenges.

At first, industrial processes are diverse and the associated data can be stationary, cyclic and/or including random variabilities. For each process, multiple types of anomalies can be observed such as drops, drifts or time-related desynchronizations. Additionally, there can be multiple sensors, making the anomaly detection a multi-variate problem for which correlations between the sensors can be present or not.

Second, the current trend is to monitor the production processes continuously and to detect abnormal behaviour in real-time. Therefore, the systems need a high level of reactivity on a stream of heterogeneous data. Such a requirement increases the computational demand, the data storage and the number of actionable algorithms to tackle the problem at hand (see application cases in section 1).

Third, the quantity of examples of anomalies is usually low, while the quantity of data related to normal behaviour is large. The rarity of anomalies makes it especially difficult when it comes to building detection rules and a large deal of a priori human knowledge on the production process is usually needed [3, 17].

Now, we can reasonably advocate that Machine Learning could offer solutions to the issues described above. The data-driven approach can be self-adaptive, it has a high dimensional learning capability and is, by nature, deployable on many different application cases (i.e. machine types). Unsupervised methods can typically be applied to model the normal behaviour of a machine, and any deviation from the expected behaviour predicted by the model could be considered as an anomaly [23]. Supervised methods could then be applied as soon as enough examples of anomalies are available, potentially using incremental learning on top of the unsupervised model [6].

3. Inclusion of Machine Learning in Industrial Processes

Successful inclusion of machine learning into an industrial process implies that the three challenges from Section 2 are addressed. In this section, the three challenges are discussed taking into account the technical requirements of a machine learning application.

The first challenge is the data diversity which is due to the diversity itself of the underlying industrial processes. Machine Learning is by nature versatile through the learning process which reduces the effort to handle this diversity. Recent advances in the field of Deep Learning also require less and less human intervention, e.g. with the capacity to learn the feature extraction part out of raw data. Another aspect of Machine Learning is in its capacity to model multivariate data, finding correlation in a large number of dimensions where classical statistical methods are typically failing.

The second challenge refers to the *3 Vs* nature of data-intensive applications, so-called big data: volume, variety and velocity [13]. As in the last decade, many software applications have been created to address the aforementioned dimension, today's available solutions benefit from a great level of maturity like e.g. Apache Kafka as a distributed stream platform or Apache Cassandra as a distributed database. Recent Machine Learning models are also based on so-called *computational graphs* with highly *parallelizable* architectures that fit well on dedicated computational hardware such as GPUs. Libraries such as Tensorflow or PyTorch are exploiting such features to handle the intensity of the data to process.

The third challenge is about the rarity of anomalies, which makes the application of Machine Learning counter-intuitive as, by nature, many examples are needed for learning. Typical approaches are *indirect*, attempting to model the normality of the process where the quantity of data is large. Anomalies are then detected when the observed data deviate from the predictions done by machine learning. Whenever few or many examples of anomalies are becoming available, classical supervised approaches can be used. The collection of examples of anomalies is also non-trivial as it requires long-term storage of very large quantities of historical data. Depending on the frequency of anomalous incidents and depending on the availability of verified examples of anomalies, different machine learning approaches could be used. A formalisation of the applicable learning concepts is summarized in Table 1 that gives a typology to the industrial machine learning settings [21].

Table 1. Typology of industrial anomaly detection settings in association with Machine Learning strategies

	Type A	**Type B**	**Type C**
Number of anomalies	None	Few	Many
Machine learning category	Unsupervised	Semi-supervised / Active learning	Supervised
Learning Concept	Deviation from normality	Discrimination of normality and abnormality	Classification of anomalies

4. Architecture

Classical supervision and anomaly detection systems were initially designed to detect simple events such as machine downtime or threshold overrun on a given sensor. As a consequence, classical system architectures are rather simple with, for example, short-term data storage and small processing capacity. Quite often, the detection is embedded in the machine itself and accessing sensor data or parameter settings can be challenging.

Here, we advocate for a de-centralized scalable system architecture meeting the prerequisites of being adaptable, maintainable and accommodating to the needs of ML [11] [22]. Similar to a notification system in healthcare [8], independent services need to be connected to collect data from sensor and trigger alarms when needed. Thus, we propose an architecture for the surveillance of machine health illustrated in Figure 1.

The proposed architecture is split into three layers with different concerns. The first is the Industrial Layer. It aims at the sensing, collection and storage of all industrial data. The second is the ICT-Service layer and acts as the data processing unit running software services such as event detection or machine learning models. The last layer is the User Layer. It is responsible for the provision of client interfaces targeting human and machine to interact with the whole system.

In the following sections, we provide further details about each block of the architecture, explaining their objectives in the context of industrial process monitoring.

4.1. Industrial Layer

Block 1 represents the industrial machinery from which sensor data and setting parameters are observed. Continuous streams of data are typically produced by potentially a large number of machines located in industrial shop floors. The streams of data can then be utilized by the alarm system.

Block 2.1 and 2.2 show sources of other contextual data. For instance, the manufacturing execution system (MES) or enterprise resource planning (ERP) system that could be used to augment the raw sensor data produced in block 1.

Block 3 is an industry 4.0 gateway able to transfer data between the machines, the alarm system and the data store. Such gateways typically allow to use multiple communication protocols and act as a buffer for a short-time window of historical data.

Block 4 represents the long-term storage database that will be used to create machine learning models. The architecture of such a system should be remote, de-coupled and based on Big Data technologies. This enables the storage to be scalable from one to multiple industrial machines [16].

4.2. ICT Services Layer

Block 5 acts as the detection agent tasked with finding specific data pattern that we will call events. This block is comprised of one or more microservices, each tasked with a singular event detection routine. These microservices need to be developed according to the requirements of the project at hand and could be as simple as detecting the login of a user or the data crossing over a threshold.

Block 6 is a machine learning based event detection system. It deals with the detection of machine learned patterns within the data.

Block 7 is the active learning feedback loop where the user interacts with the system in two different ways. First, with the rule induction system where the user verifies or discards learned rules. Second, with the training of the models based on the feedback given on the detected anomalies/events.

Block 8 is the rule induction system, tasked with finding patterns in the collected sequence of events. This leads to the discovery of new, more complex rules.

Block 9 is the rule-based decision engine. It takes the decision whether an event or a set of events triggers an alarm based on user-defined or machine induced rules. If the rule engine deems a rule to be violated, an alarm can be triggered.

Block 10 is the application that communicates alarm notifications to the clients. Also, it provides interfaces to communicate with the ICT services layer components. This application entails the management of the rules by the user. For instance, their creation, maintenance and feedback to the active learning component in block 7.

4.3. User Layer

Block 11 and 12 represent the devices and other presentation agents such as smartphones and/or web browsers.

5. Use Case Chemical Reactor

In this section, we present a partial implementation for the architecture proposed in Section 4. The use case is an industrial chemical reactor located in the chemical department of the School of Engineering and Architecture of Fribourg (see Figure 2.A). The reactor is used by students and scientific collaborators for chemical synthesis on a weekly basis. As there was a need to detect incorrect usage patterns on the platform, an automated fault detection system was commissioned.

5.1. Industrial Layer

At the first layer, the reactor (block 1) is equipped with sensors measuring the chemical synthesis carried out on a day to day basis. In total, there are thirty-nine sensors collecting measurements about electricity usage, valve openings, stirring speed and the temperature of the reactor and nine sensors that are measuring the pressure of the main valve supplying the reactor with the needed reaction components such as nitrogen, de-mineralized water or vacuum.

For the contextual data (block 2.x), we took the registration of the users and time reservation of the reactor logged by the school's badging system. This information is interpreted as contextual events by the rule engine (block 9).

The data is sent to an Industry 4.0 Gateway (block 3), with the OPC-UA communication protocol [14]. From this gateway, the collected data is forwarded via a TCP/IP connection to the blocks analysing it and to the scalable data storage and processing platform (block 4) based on an Apache Cassandra database [16, 15]

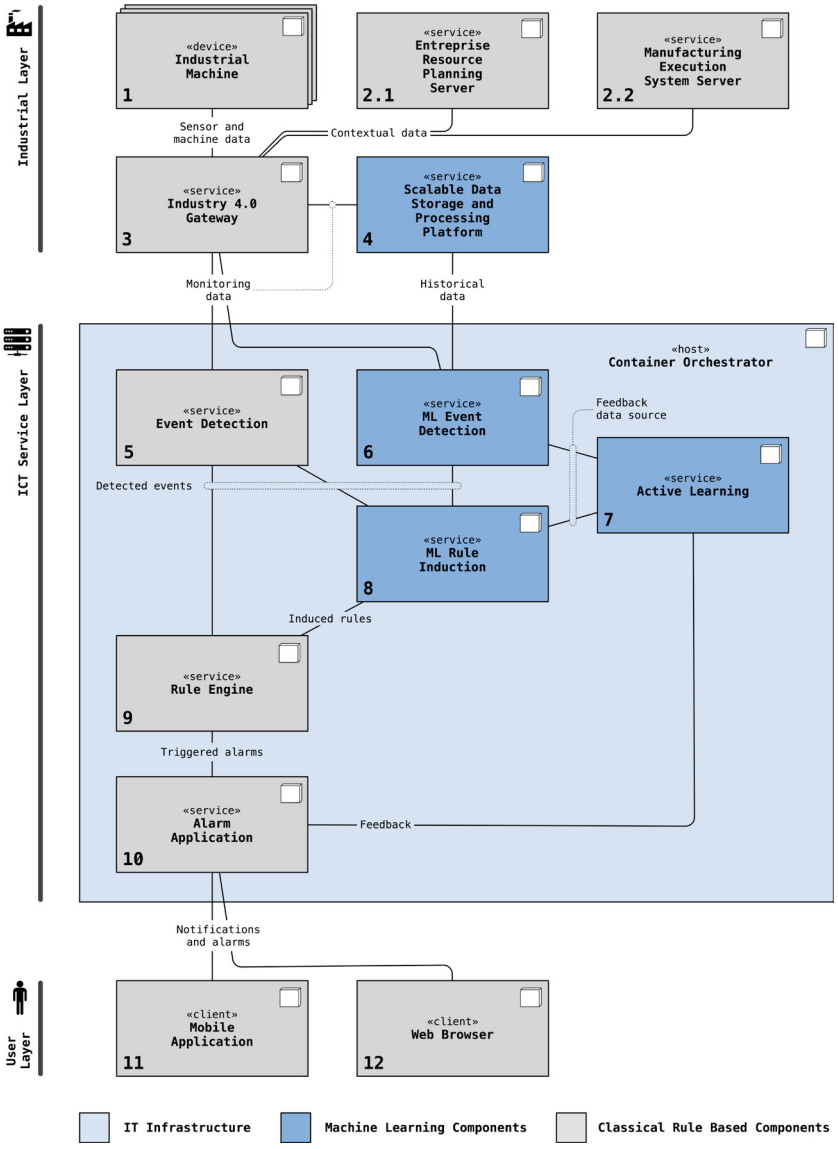

Fig. 1. Architecture of an industry 4.0 alarm system. The components are showing the information flow from sensors to the user through three different layers: Industrial, ICT Service and User. The components at the industrial layer are collecting, storing and distributing the raw data. The ICT service layer processes the information and triggers alarms if needed. Finally, at the user layer, different clients of the alarm system allows an operator to interact with the system.

5.2. ICT Service and User Layer

On this ICT service layer, we detect events based on the data coming from the gateway (block 3) with two different methods: simple threshold crossing and numerical differentiation over a sliding window. With these two applications, we can detect data fluctuations or predict when the main valve reaches pressure levels critical.

The three blocks (block 6,7,8) are envisioned to detect events and derive rules to be given to the rule engine (block 9). The algorithms are ongoing research to process heterogeneous sensor data in a generic way.

The rule engine (block 9) checks the sequences of events against a library of user-defined rules. If patterns are detected that corresponds to an entry, the rule engine triggers an alarm.

The alarm application (block 10) is developed with a Python Flask back-end. If a rule is violated an alarm message is emitted, with different severity levels. Depending on this level, a message is written to a log and/or directly sent to the responsible operator.

The front-end (block 11) is an application using the React JavaScript framework, where the operators create and manage the library of rules and logs of rule execution.

5.3. Monitoring Example and Limitations

Figure 2.B shows the main valve pressure of the nitrogen usage during a day of activity and highlights three reactor states: turned off (green), turned on (yellow) and an intermediate transition state (blue). On the plot (dark yellow), sections 1 and 2 show two regions of different usage patterns. In this region, each pressure drop represents an opening and closing of a valve by an operator.

An alarm should be triggered, when an abnormal usage pattern occurs. In areas like region 1, where the pressure is mostly stable, a threshold could be sufficient. But as the region 2 shows, the pressure can also be unstable. Therefore, a threshold system would produce many false negatives. To ensure the reliability of this system, a more elaborate and adaptive ways to detect abnormal usage pattern is needed. Classical methods, such as described in Section 5.2, are not sufficient for such a task. Therefore the inclusion of ML in the monitoring of modern industrial machinery is an inevitable step.

In Figure 2.B, a single signal of one sensor is shown, which highlights the need of ML. In reality, usage patterns are the products of different, interdependent chemical components, thus are based on multiple signals from different sensors. The detection of abnormal usage patterns then becomes a multivariate problem where machine learning methods could help to model all the non-linearities we are facing.

These two examples demonstrate why ML is necessary in the alarm detection and the validity of a modular microservice based architecture. There will be a multitude of detection tools needed to cover all different possibilities of failure in a industrial chemical reactor.

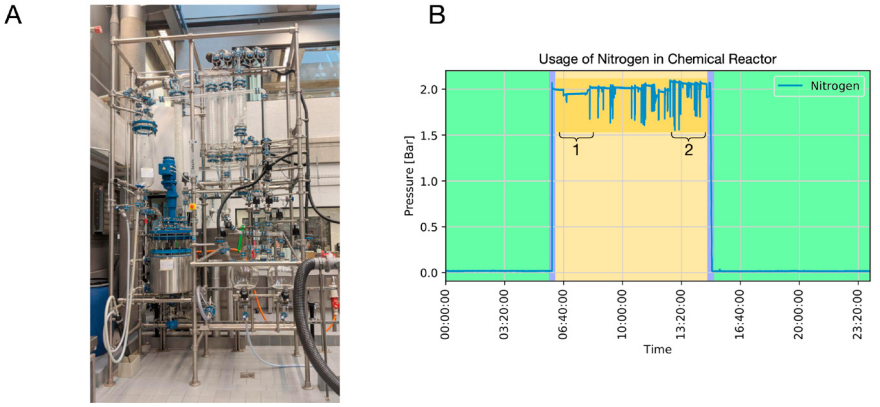

Fig. 2. A) Chemical reactor equipped with sensors; B) States of an average daily usage of nitrogen in a industrial chemical reactor. (green) Turned off reactor, (yellow) turned on reactor with spikes of usage (dark yellow) and (blue) the turning on and off of the reactor; 1) an example of an opening valve that is no closed correctly, 2) regular valve opening and closing

6. Discussion

The inclusion of ML in industrial machines is nowadays possible, as a result of the trends in IoT and Industry 4.0, but it induces some technical challenges. For instance, what modifications does a classical rule-based system need to be able to incorporate ML? How such changes influence software architecture? What are the suited algorithms to detect anomalies with machine learning for any specific real-world application and finally, what kind of technologies are needed to process these data streams in real-time?

In this study, our first result provides an approach to the monitoring of an industrial process including a machine learning work-flow. Of the three highlighted challenges, data diversity and big data nature are well understood. They refer to problems that have already been extensively discussed in applications such as Apache Spark[27, 26], Apache Kafka[10] or Apache Cassandra[12].

The rarity of anomalies is a more complex problem as model generalization is only obtainable with enough historical data. The three types of detection settings defined in section 1 provide a way to choose a ML strategy taking into account different quantities of anomalies.

One of these aforementioned strategies could be used to start the development of a ML-based AD system but shows some limitations depending on the domain and type of anomaly in section 2. Therefore, in a more advanced phase of the ML-based AD system development, multiple models should be used simultaneously. One model is responsible to detect unknown anomalies, while it is under supervision by a operator driven feedback loop. Such as a semi-supervised anomaly detection (SSAD) system will therefore improving its detection rate over time [6]. At the same time, a additional model should be responsible for AD where enough tagged anomalies can be provided.

Our second result provides a modular software architecture to updates a classical rule-based system into a machine learning one. The integration of microservices for cyber-physical systems have some documented benefits like maintainability, scalability, the delegation of team responsibilities, DevOps support, fault tolerance and separation of software responsibilities [24]. Furthermore, the use of a container orchestrator would automate the deployment, scaling and system management. The different block created in the architecture are increasing the reusability and provide good adaptability for different application cases [25].

Due to the uniqueness of industrial machinery, a custom development and individual retraining is required. In comparison with the creation of rules for each machine by hand, we expect an overall improvement in accuracy of fault detection. In the long term, such an investments should reduce the production costs of a shop floor.

Finally, our architecture provides a comprehensive way to talk about a part of a ML-based AD system and tries to go into the direction of standardization. A common naming convention to talk about the different parts of such a system would improve the communication between people. In general, we believe that the cost of implementation and maintenance could be reduced, a better interoperability reached which results in a improved detection of anomalies.

7. Conclusion

In this paper, we highlighted the potential issues of Industrial Process Monitoring (see section 2) being the diversity of processes, the reactivity of detection systems and the inherent rarity of anomalies in the day to day runtime of machines. Further, reviewed the challenges in the creation of such monitoring systems. Then we showcased that a possible tool in solving these issues is machine learning.

In Section 4, we proposed an architecture for a general alarm system that combines classical detection (e.g. threshold based) and modern, machine-learned detection of production issues. We showed the different building blocks and made a case for the additional ML-parts and why they are a necessity for a complete coverage of event detection and implementation of an alarm system.

Finally, we made a case for the architecture to be based on modular microservices. Since, this allows to adapt and reuse modules for different application cases without big ground breaking adjustments. Further, is there no general solution for the complete monitoring of a modern industrial machine and therefore combinations of different detection approaches are needed based on a case by case basis.

References

[1] Anderl, R., 2015. Industrie 4.0 -Technological approaches, use cases, and implementation. At-Automatisierungstechnik 63, 753–765. doi:10.1515/auto-2015-0025.

[2] Chalapathy, R., Chawla, S., 2019. Deep Learning for Anomaly Detection: A Survey , 1–50URL: http://arxiv.org/abs/1901.03407, arXiv:1901.03407.

[3] Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection. ACM Computing Surveys 41, 1–58. URL: http://portal.acm.org/citation.cfm?doid=1541880.1541882, doi:10.1145/1541880.1541882, arXiv:arXiv:1011.1669v3.

[4] Cuccu, G., Danafar, S., Cudre-Mauroux, P., Gassner, M., Bernero, S., Kryszczuk, K., 2017. A data-driven approach to predict NOx-emissions of gas turbines. Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017 2018-Janua, 1283–1288. doi:10.1109/BigData.2017.8258056.

[5] Goh, J., Adepu, S., Tan, M., Lee, Z.S., 2017. Anomaly detection in cyber physical systems using recurrent neural networks. Proceedings of IEEE International Symposium on High Assurance Systems Engineering , 140–145doi:10.1109/HASE.2017.36.

[6] Görnitz, N., Kloft, M., Rieck, K., Brefeld, U., 2013. Toward supervised anomaly detection. Journal of Artificial Intelligence Research 46, 235–262. doi:10.1613/jair.3623, arXiv:1401.6424.

[7] Heiner Lasi. Hans Georg Kemper. Peter Feltke. Thomas Feld. Michael Hoffmann, 2014. Indusrtry 4.0. Business& Information System Engineering 6, 239–242. URL: https://link.springer.com/article/10.1007{%}2Fs12599-014-0334-4, doi:10.1007/s12599-014-0334-4.

[8] Hill, R., Shadija, D., Rezai, M., 2018. Enabling community health care with microservices. Proceedings - 15th IEEE International Symposium on Parallel and Distributed Processing with Applications and 16th IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC 2017 , 1444–1450doi:10.1109/ISPA/IUCC.2017.00220, arXiv:1709.07037.

[9] Kanawaday, A., Sane, A., 2018. Machine learning for predictive maintenance of industrial machines using IoT sensor data. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2017-Novem, 87–90. doi:10.1109/ICSESS.2017.8342870.

[10] Kreps, J., Narkhede, N., Rao, J., 2011. Kafka: a Distributed Messaging System for Log Processing. ACM SIGMOD Workshop on Networking Meets Databases , 6URL: http://research.microsoft.com/en-us/um/people/srikanth/netdb11/netdb11papers/netdb11-final12.pdf.

[11] Krylovskiy, A., Jahn, M., Patti, E., 2015. Designing a Smart City Internet of Things Platform with Microservice Architecture. Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015 , 25–30doi:10.1109/FiCloud.2015.55.

[12] Lakshman, A., Malik, P., 2010. Cassandra: A decentralized structured storage system. SIGOPS Oper. Syst. Rev. 44, 35–40. URL: http://doi.acm.org/10.1145/1773912.1773922, doi:10.1145/1773912.1773922.

[13] Lee, I., 2017. Big data: Dimensions, evolution, impacts, and challenges. Business Horizons 60, 293–303. URL: http://dx.doi.org/10.1016/j.bushor.2017.01.004, doi:10.1016/j.bushor.2017.01.004.

[14] Leitner, S.H., Mahnke, W., 2006. OPC UA Service-oriented Architecture for Industrial Applications. Softwaretechnik-Trends 26, 1 – 6. URL: http://www2.cs.uni-paderborn.de/cs/ag-engels/GI/ORA2006-Papers/leitner-final.pdf{%}5Cnhttp://dblp.uni-trier.de/db/journals/stt/stt26.html{#}LeitnerM06{%}5Cnhttp://www2.cs.uni-paderborn.de/cs/ag-engels/GI/ORA2006-Papers/leitner-final.pdf.

[15] Linder, L., Hennebert, J., Esseiva, J., 2018. Bbdata, a big data platform for smart buildings, in: FTAL conference on Industrial Applied Data Science, pp. 38–39.

[16] Linder, L., Vionnet, D., Bacher, J.P., Hennebert, J., 2017. Big Building Data-a Big Data Platform for Smart Buildings. Energy Procedia 122, 589–594. URL: https://doi.org/10.1016/j.egypro.2017.07.354, doi:10.1016/j.egypro.2017.07.354.

[17] Mahdavinejad, M.S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A.P., 2018. Machine learning for internet of things data analysis: a survey. Digital Communications and Networks 4, 161–175. URL: https://doi.org/10.1016/j.dcan.2017.10.002, doi:10.1016/j.dcan.2017.10.002, arXiv:1802.06305.

[18] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G., 2016. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection URL: http://arxiv.org/abs/1607.00148, arXiv:1607.00148.

[19] Martí, L., Sanchez-Pi, N., Molina, J.M., Garcia, A.C.B., 2015. Anomaly detection based on sensor data in petroleum industry applications. Sensors (Switzerland) 15, 2774–2797. doi:10.3390/s150202774.

[20] Onal, A.C., Berat Sezer, O., Ozbayoglu, M., Dogdu, E., 2017. Weather data analysis and sensor fault detection using an extended IoT framework with semantics, big data, and machine learning. Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017 2018-Janua, 2037–2046. doi:10.1109/BigData.2017.8258150.

[21] Rychener, L., Esseiva, J., Hennebert, J., 2018. Machine Learning for Anomaly Detection in Time-Series Produced by Industrial Processes. Proceedings - FTAL conference on Industrial Applied Data Science 1, 7–8.

[22] Salikhov, D., Khanda, K., Gusmanov, K., Mazzara, M., Mavridis, N., 2017. Microservice-based IoT for Smart Buildings. Proceedings - 31st IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2017 , 302–308doi:10.1109/WAINA.2017.77, arXiv:1610.09480.

[23] Stojanovic, L., Dinic, M., Stojanovic, N., Stojadinovic, A., 2016. Big-data-driven anomaly detection in industry (4.0): An approach and a case study. Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016 , 1647–1652doi:10.1109/BigData.2016.7840777.

[24] Taibi, D., Lenarduzzi, V., Pahl, C., 2017. Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation. IEEE Cloud Computing 4, 22–32. doi:10.1109/MCC.2017.4250931.

[25] Thramboulidis, K., Vachtsevanou, D.C., Solanos, A., 2018. Cyber-physical microservices: An IoT-based framework for manufacturing systems. Proceedings - 2018 IEEE Industrial Cyber-Physical Systems, ICPS 2018 , 232–239doi:10.1109/ICPHYS.2018.8387665, arXiv:1801.10340.

[26] Zaharia, M., Chowdhury, M., Das, T., Dave, A., 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation , 2–2URL: https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf, doi:10.1111/j.1095-8649.2005.00662.x, arXiv:EECS-2011-82.

[27] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I., 2010. Spark: Cluster computing with working sets. HotCloud 10, 95.